代数方程式の数値解法に関するJenkins-Traubの方法について (数値解析の基礎理論研究会報告集)

AUTHOR(S):

池辺, 八洲彦

CITATION:

池辺, 八洲彦. 代数方程式の数値解法に関するJenkins-Traubの方法について (数値解析の基礎理論研究会報告集). 数理解析研究所講究録 1969, 72: 51-68

ISSUE DATE:

1969-05

URL:

http://hdl.handle.net/2433/107942

数理解析研究所講究録
72巻 1969年 51-68

代数方程式の数値解法に関する Jenkins-Traub の方法について

京大 工 池辺八洲彦

代数方程式の根の数値解法として、最近、Jenkins および Traub が発表した方法[1]は注目に値すると思われるので、ここに概略を紹介し、ついで、京都大学大型計算機センター FACOM-230-60 を用いて行った数値計算結果をつけくわえることにする。この方法は Iterative Process であるが、

(a) 根の多重性は收束のさまたげにならない

(b) つねに收束する

の2点を特徴とする。また、以下の説明からわかるように、アルゴリズムも簡単である。

文献[1]で指摘されているとおり、この方法は、実は、与えられた代数方程式をとくという問題を、行列の固有値問題に還元している、すなわち、その代数方程式のコンパニオン行列に Inverse Power Method を適用したものとみなせるのである。

52

1. 定義

$$P(z) = z^n + a_1 z^{n-1} + \cdots + a_n \quad (a_n \neq 0)$$

を与えられた複素係数の多項式とし、$P(z)=0$ の一根を求めることを考える。 いま、$\rho_1, \cdots \rho_m$ を、たがいにことなる $P(z)=0$ の根とし、それぞれの多重度を l_i $(i=1, \cdots, m)$ とする。 すなわち、

$$P(z) = (z-\rho_1)^{l_1} \cdots (z-\rho_m)^{l_m}$$

とかけることになる。 また、

$$P_i(z) = P(z) / (z-\rho_i) \quad (i=1, \cdots m)$$

とおく。 いま、s を $P(z)=0$ の根ではない一つの複素数とするとき、$n-1$ 次の多項式を $n-1$ 次の多項式に map する次のような linear operator T_s を考える：

$$T_s H(z) = \frac{1}{z-s} \left\{ H(z) - \frac{H(s)}{P(s)} P(z) \right\}$$

剰余の定理により、右辺の括弧 $\{\cdot\}$ 内の多項式は $z-s$ でわりきれるから、右辺は、たかだか $n-1$ 次の多項式 $H(z)$ に対して、たしかに、$n-1$ 次の多項式を与える。

以下の議論において、多項式 $P(z)$ はつねに一つの固定された多項式として考える。

2. 定理

$$T_s P_i(z) = \frac{1}{\rho_i - s} P_i(z) \quad (i = 1, \cdots, m)$$

すなわち、$(\rho_i - s)^{-1}$ は operator T_s の固有値であり、$P_i(z)$ が、それに対応する固有ベクトルである。

証明　T_s の定義式に直接代入することによって検証できる。

3. 定理 (Fixed-Shift Iteration)

$$H_0(z) = d P(z)/dz$$

$$H_{k+1}(z) = T_s H_k(z) \quad (k = 0, 1, 2, \cdots)$$

なる iteration を考える。　すると、

$$H_k(z) = l_1 \frac{P_1(z)}{(\rho_1 - s)^k} + \cdots + l_m \frac{P_m(z)}{(\rho_m - s)^k} \quad (k = 0, 1, \cdots)$$

もし、$|\rho_1 - s| < |\rho_i - s| \ (i \neq 1)$ ならば、任意の z に対して

$$z - P(z)/\tilde{H}_k(z) \longrightarrow \rho_1 \quad (k \to \infty)$$

ただし、$\tilde{H}_k(z)$ は $H_k(z)$ を "normalize" したもの、すなわち、最高次の係数で割ったものを意味する。

証明　前半の証明は前定理より出る。　後半については、$k \to \infty$ のとき、$H_k(z) \rightrightarrows \cdots$ なる式の右辺は漸近的に、$\rho_1 P_1(z)/(\rho_1 - s)^k$ とみなしてよい。　すなわち、$k \to \infty$

のとき、$\tilde{H}_k(z)$ は漸近的に $P_1(z)$ とみなせる。 これより定理の後半が証明できる。

4. 注意

前節の收束は $|\beta_1 - s| / |\beta_i - s|$ $(i = 2, \cdots, n)$ なる比によって速度が左右される。 すなわち、s が β_1 に近いほど收束がはやくなる。 一方、前定理により $s - P(s)/\tilde{H}_k(s) \to \beta_1$ だから、ある程度、收束がすすんだら、s として、つねに同じ s を用いないで、β_1 により近い s を用いることが考えられる。 このことより、つぎの Variable-shift Iteration が考えられる。 すなわち、

$$H_{k+1}(z) = \frac{1}{z - s_k}\left\{H_k(z) - \frac{H_k(s_k)}{P(s_k)}P(z)\right\}$$

$$s_{k+1} = s_k - \frac{P(s_k)}{\tilde{H}_{k+1}(s_k)}$$

$k \geq L$, L は十分大きい整数

5. Jenkins および Traub の提唱した方法

Jenkins と Traub は、つぎの3つの stage よりなる方法を提唱した。

<u>Stage 1</u> 与えられた $P(z)$ に対して、$s = 0$ とおいて、定理3でのべた Fixed-shift Iteration を M 回行う。 J.-T.

は $M=5$ とおいている。

Stage 2　つぎに、s に一つの適当な値を与えて、Fixed-Shift Iteration を適当回行う。この場合、s としては、$P(z)=0$ の絶対値の最小の根になるべく近いようにするために、$P(z)=0$ の根の lower bound を一つ求める。それには、

$$z^n + |a_1| z^{n-1} + \cdots + |a_{n-1}| \cdot z - |a_n| = 0$$

なる方程式の（unique な）正の根を求め、これを β とおけば、β は $P(z)=0$ の根のどれよりも絶対値が大きくない。β の求め方としては、Newton の方法がよい。さてそこで

・s として $|s|=\beta$ なるような s を勝手に一つとる。J.-T. は、相隣る3つの k の値について $t_k \equiv s - P(s)/\tilde{H}_k(s)$ なる値を計算し、$|t_{k+1}-t_k| \leq |t_k|/2$ かつ $|t_{k+2}-t_{k+1}| \leq |t_{k+1}|/2$ なる条件が満足されれば、ここで Stage 2 を打切るという。これらの条件が"いつまでも"満足されない場合は、s のとり方をかえるのである。

Stage 3　前節で説明した Variable-shift Iteration を実行する。収束しない場合には、ふたたび、Stage 2 にもどる。J.-T. によれば、収束しない場合は稀で、その場合でも、Stage 2 の s の値として、3つ以上試してみる場合は経験されなかったという。

以上の操作により、$P(z)=0$ の根のうち、絶対値の最小のものが求まりやすい。 そうでなくても、Stage 2 のうつり方によって、Stage 3 を終了して求められた根は、$P(z)=0$ の絶対値最小の根にくらべて、高々、3倍の大きさをもつにすぎない（これは三角不等式だけを用いて容易に証明できる）。 これは $P(z)=0$ の根をすべて求めようとする場合、精度の観点からきわめてのぞましい（[1, p.2]）。

6. 定理（收束定理）

前節の J.-T. Three Stage Iteration を考える。 4節の記法を用いて、$k=L$ のときより Stage 3 に入るものとする。すると、以下の条件(i)～(iv) が真ならば、J.-T. 法は收束する、すなわち $S_k \longrightarrow \rho_1$: (i) $P(S_L) \neq 0$; (ii) $|S_L - \rho_1| < R/2$, $R = \min\{|\rho_i - \rho_1| : i = 2, \cdots, m\}$; (iii) $h_1^{(L)} \neq 0$; (iv) $\sum_{i \neq 1} |h_i^{(L)}| < |h_1^{(L)}|/3$ 、ただし、$H_k(z) = h_1^{(k)} P_1(z) + \cdots + h_m^{(k)} P_m(z)$ に従って、係数 $h_1^{(k)}$, $\cdots$ を定義している。

証明に先立ち、$D_L = \sum_{i \neq 1} |h_i^{(L)}| / |h_1^{(L)}|$ とおく。すると、上の条件(iv) は $D_L < 1/3$ とかいてよい。

証明 以上の条件のもとで、Iteration が well-defined であることの証明はあとまわしにして、收束性の方を先に証

明する。　まず、初歩的な代数演算をいくらか実行すると、次式が得られる。

$$\frac{s_{k+1}-\rho_1}{s_k-\rho_1}=\frac{\sum_{i\neq 1}\left[r_i^{(k)}\right]^2 d_i^{(k)}-\sum_{i\neq 1} r_i^{(k)} d_i^{(k)}}{1+\sum_{i\neq 1} r_i^{(k)} d_i^{(k)}} \qquad (*)$$

ただし、$r_i^{(k)}=(\rho_1-s_k)/(\rho_i-s_k)$、$d_i^{(k)}=h_i^{(k)}/h_1^{(k)}$　$(i\neq 1)$

さて、$s_k\to\rho_1$ なることを示すには、$|s_{k+1}-\rho_1|/|s_k-\rho_1|\leq A<1$ なるごとき、k に依存しないような定数 A が存在することをいえば十分である。　そのような定数 A として、$2D_L/(1-D_L)$ をとってよいことを示そう。　仮定 (iv) により、この数が厳密に 1 より小さい正数であることは明らかである。　つぎに、(*) 式において、まず、$k=L$ の場合を考える。　s_L が (ii) の意味で ρ_1 にもっとも近いという事実より、$|r_i^{(L)}|<1$ $(i\neq 1)$、したがって、(*) 式より、$|s_{L+1}-\rho_1|/|s_L-\rho_1|$ なる比が、$2D_L/(1-D_L)$ によっておさえられることがわかる。この値は 1 より小さいから、仮定 (ii) より、$|s_{L+1}-\rho_1|<R/2$、それゆえ、$|r_i^{(L+1)}|<1$ をうる。　また、$d_i^{(L+1)}$ について考えると、$d_i^{(L+1)}=d_i^{(L)}\cdot r_i^{(L)}$　$(i\neq 1)$ だから、$|d_i^{(L+1)}|<|d_i^{(L)}|$、それゆえ、(*) 式において、$k=L+1$ の場合を考えると、簡単な議論で $|s_{L+2}-\rho_1|/|s_{L+1}-\rho_1|\leq 2D_L/(1-D_L)<1$ が得られる。　以下同様の議論をくりかえすことにより、上にあげた命題を証明できる。

つぎに、数列 $\{s_k\}_{k=L}^{\infty}$ が well-define できることを示すには、$\widehat{H}_{k+1}(s_k)$, $k=L, \cdots$ が 0 にならないことをいえばよい。

そのために、上と同様な計算により、

$$\widehat{H}_{k+1}(s_k) = \frac{1 + \sum_{i\neq 1} d_i^{(k)} \left[r_i^{(k)}\right]^2}{1 + \sum_{i\neq 1} d_i^{(k)} r_i^{(k)}} \cdot P_1(s_k), \quad k=L, L+1, \cdots$$

をうる ($P_1(z) = P(z)/(z-\beta_L)$)。 $P_1(s_k) \neq 0$ (仮定(i))だから、また上の結果より、$\sum_{i\neq 1} |d_i^{(k)}| < 1/3$ ⇒ $|r_i^{(k)}| < 1$ $(i\neq 1)$ だから、$\widehat{H}_{k+1}(s_k)$, $k=L, \cdots$, は 0 になることはない。

7. 定理(大域収束定理)

J.-T. 法において、Stage 1 を任意回実行し、Stage 2 を十分長く実行したのちに、Stage 3 に移行すれば、J.-T. 法は必ず収束する。ただし、Stage 2 におけるシフトパラメーター s は $P(z)=0$ の根(相異る根) $\beta_1, \cdots, \beta_m$ のうちのどれか1つだけにもっとも近いものとする。(J.-T. 法はこの根に収束する)。

証明　Stage 2 のパラメーター s は根 β_1 にもっとも近いものとする。今までの notation を踏襲し、stage 1 を M 回、Stage 2 を L−M 回実行したのちに、Stage 3 に入るものとする。すると、$H_0(z) = \sum_{i=1}^{m} l_i \cdot P_i(z) = P'(z)$ より出発して、次式をうる。

$$H_L(z) = \sum_{i=1}^{m} \rho_i^{-M} (\rho_i - s)^{-(L-M)} \ell_i \quad P_i(z)$$
$$\equiv \sum_{i=1}^{m} h_i^{(L)} P_i(z)$$

定理6の D_L を計算すると、

$$D_L = \sum_{i\neq 1} |h_i^{(L)}| / |h_1^{(L)}| = \sum_{i\neq 1} \left|\frac{\rho_1}{\rho_i}\right|^M \left|\frac{\rho_1 - s}{\rho_i - s}\right|^{L-M} \frac{\ell_i}{\ell_1}$$

M を固定し、L を十分大きくとれば、$|\rho_1 - s| < |\rho_i - s|$ $(i \neq 1)$ だから、D_L はいくらでも小さくとれる。そこで、$D_L < 1/3$、かつ、$|\rho_1 - s| \frac{2D_L}{1-D_L} < R/2$, $R = \min_{i\neq 1} |\rho_1 - \rho_i|$ なるごとくにとる。すると、前定理の証明の中で用いた議論と同様の議論で、$|\rho_1 - S_L| < R/2$ となる、但し、$S_L = s - P(s)/\widetilde{H}_L(s)$。これで前定理の仮定はすべて満足されたから、$S_k \to \rho_1$ なることが結論できる。

8. 定理（収束速度に関する定理）

定理6の仮定が真ならば、$k > L$ に対して、

$$|S_{k+1} - \rho_1| \leq \varepsilon_k |S_k - \rho_1|^2$$
$$\varepsilon_k = \frac{2}{R}\left(\frac{2D_L}{1-D_L}\right)^{(k-L)(k-L-1)/2}$$

証明　定理6の証明で用いたのと同種の論法を用いる。

定理6の証明の(*)式より、

$$\frac{S_{k+1} - \rho_1}{(S_k - \rho_1)^2} = \frac{\sum_{i\neq 1} \frac{r_i^{(k)} d_i^{(k)}}{S_k - \rho_i} - \sum_{i\neq 1} \frac{d_i^{(k)}}{S_k - \rho_i}}{1 + \sum_{i\neq 1} \left[r_i^{(k)}\right]^2 d_i^{(k)}}$$

をうる。ここで、

$$r_i^{(k)} = \frac{s_k - \beta_1}{s_k - \beta_i} = \frac{s_k - \beta_1}{s_{k-1} - \beta_1} \cdot \frac{s_{k-1} - \beta_1}{s_{k-2} - \beta_1} \cdots \frac{s_{L+1} - \beta_1}{s_L - \beta_1} \cdot \frac{s_L - \beta_1}{s_k - \beta_i}$$

これより、$|r_i^{(k)}| \leq \left(\frac{2D_L}{1-D_L}\right)^{k-L}$ をうる。 また、
$|s_k - \beta_i| > \frac{R}{2}$ $(i \neq 1)$, $d_i^{(k)} = r_i^{(k-1)} d_i^{(k-1)} = \cdots$ だから、
$|d_i^{(k)}| \leq \left(\frac{2D_L}{1-D_L}\right)^{(k-L)(k-L-1)/2}$ をうる。 これらを、頭初の式の右辺に考慮すれば、求める結果をうる。

9. 注意

今までは、おもに、Algorithmを中心にのべてきたが、この方法は、行列の固有値問題を Inverse Power Method でといていることに帰着していることを示そう。 J.-T. 法の中心となるのは、T_s という operator であるが、これは第1節でのべたとおり、$n-1$ 次の多項式をそれ自身の上に写す線形作用素である。 以下、$\mathcal{P}_{n-1}$ をもって、$n-1$ 次の多項式全体の集合をあらわすことにする。

10. 定理

Stage 1 にあらわれる "No Shift Operator" T_0、すなわち、

$$T_0 f(z) = \frac{1}{z}\left\{f(z) - \frac{f(0)}{P(0)} \cdot P(z)\right\}, \quad f \in \mathcal{P}_{n-1}$$

を考えると、この operator は $\{z^{n-1}, z^{n-2}, \cdots, z, 1\}$ なる ordered base に関して、つぎの行列表現をとる、但し、

$$P(z) = z^n + a_1 z^{n-1} + \cdots + a_n , \qquad a_n \neq 0 .$$

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & -1/a_n \\ 1 & 0 & \cdots & 0 & -a_1/a_n \\ 0 & 1 & \cdots & 0 & -a_2/a_n \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & \cdots & 1 & -a_{n-1}/a_n \end{bmatrix}$$

T_0 の固有値は $1/\rho_i$ $(i=1,\cdots,m)$ (多重度 l_i) である、ただし、いままでどおり、$\rho_1,\cdots,\rho_m$ はそれぞれ多重度 $l_1,\cdots,l_m$ の $P(z)=0$ の根である。 また、$P_i(z) = P(z)/(z-\rho_i)$ は $1/\rho_i$ に応ずる固有ベクトルである。

証明　前半は $f(z)$ として、順次、$z^{n-1}, z^{n-2}, \cdots, z, 1$ とおいて、$T_0 f(z)$ がどうなるかをみればよい。 たとえば、$T_0(z^{n-1}) = z^{n-2}$。 $T_0 P_i(z) = (1/\rho_i) P_i(z)$ なることもすでに定理2でのべた。 上の行列の固有多項式をとれば、それは $P(z)=0$ の根の逆数を根とする多項式にひとしいことがわかる。

(注) 上の行列は $P(z)=0$ において、$z = 1/\zeta$ とおいた、ζ に関する多項式のコンパニオン行列になっている。

11. 定理

$$T_s = (T_0^{-1} - sI)^{-1}$$

証明　T_s の逆を求めると、それは存在して、

$$T_s^{-1} g(z) = (z-s)\, g(z) - b_0\, P(z) \qquad g(z) \in \mathcal{P}_{n-1}$$

ただし、b_0 は $g(z)$ の最高次（$n-1$ 次）の項の係数である。

上式を変形すれば、

$$\begin{aligned} T_s^{-1} g(z) &= \left[z \cdot g(z) - b_0 P(z) \right] - s\, g(z) \\ &= T_0^{-1} g(z) - s\, g(z) \\ &= \left(T_0^{-1} - s\, I \right) g(z) \end{aligned}$$

これより証明すべき関係式をうる。

12. 注意

Operator T_0^{-1} の行列表現を求めると、それは、

$$\begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ -a_n & 0 & 0 & \cdots & 0 \end{bmatrix}$$

これは与えられた多項式 $P(z) = z^n + a_1 z^{n-1} + \cdots$ に対するコンパニオン行列に他ならない。この行列の固有多項式は $P(z)$ と一致する。

前定理によって J.-T.法の iterative kernel T_s の構造がわかるが、上の T_0^{-1} の行列表現を考慮すると、すでにのべたとおり、J.-T.法が $P(z)$ のコンパニオン行列に *inverse power iteration* を適用したものであることがわかる。しかし、このことは

つぎの定理によって明確となる。

13. 定理

Stage 3 の Variable-Shift Iteration

$$H_{k+1}(z) = T_{s_k} H_k(z) = \frac{1}{z - s_k}\left[H_k(z) - \frac{H_k(s_k)}{P(s_k)} P(z) \right]$$

$$s_{k+1} = s_k - \frac{P(s_k)}{\tilde{H}_{k+1}(s_k)}$$

はつぎの matrix iteration と同値である：

$$\underline{h}^{(k+1)} = (A - s_k I)^{-1} \underline{h}^{(k)}$$

$$s_{k+1} = \left[\underline{s}^{(k)}\right]^T A \cdot \underline{h}^{(k+1)} \Big/ \left[\underline{s}^{(k)}\right]^T \underline{h}^{(k+1)}$$

ただし、

$$\left[\underline{s}_k\right]^T = \left[1, s_k, s_k^2, \cdots, s_k^{n-1}\right] \qquad (T \text{は転置を示す})$$

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_n \\ 1 & 0 & \cdots & 0 & -a_{n-1} \\ 0 & 1 & \cdots & 0 & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & \cdots & 1 & -a_1 \end{bmatrix}$$

$$\left[\underline{h}^{(k)}\right]^T = \left[\alpha_n^{(k)}, \alpha_{n-1}^{(k)}, \cdots, \alpha_1^{(k)}\right]$$

$$H_k(z) \equiv \alpha_1^{(k)} z^{n-1} + \alpha_2^{(k)} z^{n-2} + \cdots + \alpha_n^{(k)}$$

証明　$H_{k+1}(z) = \cdots$ なる式と $\underline{h}^{(k+1)} = \cdots$ なる式とが同値であることは、$z^{n-1}, \cdots, z, 1$ なる test polynomials に対して、T_s および $(A - sI)^{-1}$ がどう挙動するかをみればよい。$s_{k+1} = \cdots$ なる式については簡単な計算で同値性が容易にたしか

められる。

14. 定理

Stage 3 の $s_{k+1} = s_k - P(s_k) / \widetilde{H}_{k+1}(s_k)$ なる iteration は $W(z) = P(z) / H_k(z)$ なるその関数に Newton-Raphson 法を適用したものとみなせる。

証明前に一言すれば、k が十分大きければ、$H_k(z)$ は漸近的に $P(z)/(z-\zeta_1)$ に比例している。ただし、$s_k \to \zeta_1$ なるものとする。すると、上に定義した $W(z)$ という関数は、ほぼ $z-\zeta_1$ に比例している。この関数は一次式だから、Newton-Raphson 法を適用すれば、一度で真の根を得る筈である。これより、J.T. 法の收束速度のよいことがうかがわれる。実際、定理8よりわかるように、收束は、ほぼ、quadratic である。

証明　$z = s_k$ を初期の近似値として、$W(z)$ に Newton-Raphson 法を適用し、新しい近似値を t とおけば、

$$t = s_k - P(s_k) H_k(s_k) / \left(P'(s_k) H_k(s_k) - P(s_k) H_k'(s_k) \right)$$

他方、$\widetilde{H}_{k+1}(z) = \dfrac{1}{z - s_k} \left[P(z) - \dfrac{P(s_k)}{H_k(s_k)} H_k(z) \right]$ だから、

$\widetilde{H}_{k+1}(s_k) = \lim_{z \to s_k} \widetilde{H}_{k+1}(z) = P'(s_k) - P(s_k) H_k'(s_k) / H_k(s_k)$。これを $s_{k+1} = s_k - \cdots$ なる式の右辺に代入して計算すれば、上に計算した t と、s_{k+1} とが一致することがわかる。

15. 数値実験例

J.-T.法のessentialな部分をためすために、次例の方程式の根を求めることにした。手続きとしては、stage 1のiterationを5回くりかえし、stage 2を省略していきなりstage 3に入った。次例で $\varepsilon =\cdots$ とある値がstage 3に入ったときの根の近似値である。ε は収束を判定する値で、$|s_{k+1}-s_k|/|s_{k+1}|<\varepsilon$ がみたされたとき、s_{k+1} をもって答(根)とする。すでにのべたごとく、使用計算機は京都大学大型計算機センターのFACOM 230-60で、使用言語はFortran(Cタイプ)である。この計算機の浮動小数点単精度演算の精度は26ビット、同倍精度では61ビットである(語長は1語36ビットである)。また、次例において答のそばに括弧でつつんで示してある数字は、stage 3のiterationが行われた回数である。

例1は3重根が1個、2重根が1個ある場合であるが、これは次例の単根ばかりの場合にくらべて、当然予想されるとおり、収束もおそく、また精度も劣っている。

例1. $P(z)=(z-1)^3(z-2)^2=z^5-7z^4+19z^3-25z^2+16z-4$

		単精度		倍精度	
$S=0.25$	$\varepsilon=10^{-6}$	1.008547		1.000000	(24)
	$\varepsilon=10^{-7}$	同上	(32)	0.9999974	(33)
$S=0.75$	$\varepsilon=10^{-6}$	1.007635		0.9999999	(31)
	$\varepsilon=10^{-7}$	同上	(12)	0.9999966	(44)
$S=1.25$	$\varepsilon=10^{-6}$	1.008994		1.000002	(19)
	$\varepsilon=10^{-7}$	同上	(19)	1.000002	(23)
$S=1.75$	$\varepsilon=10^{-6}$	1.998136		1.999999	(14)
	$\varepsilon=10^{-7}$	同上	(19)	2.000000	(16)

例2. $P(z)=(z-1)(z-2)(z-3)(z-4)(z-5)$

$=z^5-15z^4+85z^3-225z^2+274z-120$

		単精度		倍精度	
$S=0.25$	$\varepsilon=10^{-6}$	1.000000	(5)	1.000000	(5)
	$\varepsilon=10^{-7}$	〃	(5)	〃	(5)
$S=0.75$	$\varepsilon=10^{-6}$	〃	(4)	〃	(4)
	$\varepsilon=10^{-7}$	〃	(4)	〃	(4)
$S=1.25$	$\varepsilon=10^{-6}$	〃	(5)	〃	(5)
	$\varepsilon=10^{-7}$	〃	(5)	〃	(5)
$S=1.75$	$\varepsilon=10^{-6}$	1.999998	(4)	2.000000	(4)
	$\varepsilon=10^{-7}$	〃	(4)	〃	(4)

16. 結語

以上、J-T法は要するに代数方程式をとくという問題を行列の固有値問題に還元している。この場合、行列というのは、与えられた方程式に対応するコンパーオン行列であり、この行列は特別のかたち、すなわち、Hessenberg形をしている。したがって、Hessenberg形の行列に適用できる固有値の数値解法はコンパニオン行列にも適用できるから、上のJ-T法以外にも代数方程式をとく有力な方法があるものと思われる。たとえば、QR法（たとえば[2]参照）の適用は興味あるものとおもわれる。

参考文献

[1] Jenkins, M. A., Traub, J. F., "A Three-Stage Variable-Shift Iteration for Polynomial Zeros and Its Relation to Generalized Rayleigh Iteration", Technical Report No. CS107, August 26, 1968, Computer Science Department, Stanford University, Stanford California, U.S.A.

[2] Ralston, A., Wilf, H. S. (Ed.), "The LU and QR Algorithm" (by B.N. Parlett), Mathematical Methods for Digital Computers, Vol. 2, John-Wiley, New York, pp 116-130.